Simulation of the space debris environment in LEO using an analytical approach

Christopher Kebschull[a,*], Philipp Scheidemann[a], Sebastian Hesselbach[a], Jonas Radtke[a], Vitali Braun[a], H. Krag[b]

[a] *Institute of Aerospace Systems, TU Braunschweig, Hermann-Blenk-Str. 23, 38108, Braunschweig, Germany*
[b] *Space Debris Office, ESA/ESOC, Robert-Bosch-Str. 5, 64293 Darmstadt, Germany*

Abstract

Several numerical approaches exist to simulate the evolution of the space debris environment. These simulations usually rely on the propagation of a complete population of objects in order to determine the collision probability for each object. Using a Monte Carlo (MC) approach the chances for events, such as explosions and collisions, are triggered based on an assumed probability distribution. So in many different scenarios different objects are fragmented and contribute to a different version of the space debris environment. Finally, the results of the different scenarios are averaged to get a statistically significant estimation. This method is computationaly very expensive due to the propagation of the objects and the application of the MC method.
At the Institute of Aerospace Systems (ILR) an analytical model capable of describing the evolution of the space debris environment has been developed and implemented. The model is based on source and sink mechanisms, where yearly launches as well as collisions and explosions are considered as sources. The natural decay and post mission disposal measures are the only sink mechanisms. This method reduces the computational costs tremendously. In order to achieve this benefit a few simplifications have been applied. The approach of the model partitions the LEO into altitude shells. Only two kinds of objects are considered, intact bodies and fragments, which are also divided into diameter bins. As an extension to the previously presented model the eccentricity has additionally been taken into account with 67 eccentricity bins. While a set of differential equations has been implemented in a generic manner, the Euler method has been chosen to integrate the equations for a given time span. For this paper parameters have been chosen so that the model is able to reflect the results of the numerical MC-based simulation LUCA, which is also being developed at the ILR. The evolution of the population in LEO for a 200 years time span is shown and compared for both approaches using step sizes of 1 year. For selected objects in LEO the flux and environmental criticality values are shown. In conclusion the field of application for such a fast analytical model is shown.

Keywords: Evolution, Space debris environment, Analytical approach

1. Introduction

The challenging task to predict the future space debris environment has been approached using many different analytical and numerical simulations e.g. Lewis et al. (2009), Rossi et al. (1994), Radtke et al. (2013). In the following a simulation tool named SANE (**S**imple **AN**alytical **E**volution) will be presented. The model considers all relevant population influencing effects in LEO through source and sink mechanisms. They are represented as differential equations. These equations are solved by a simple Euler integrator to provide a forecast of the space debris population for any arbitrary instant in time. In order to keep the complexity of the model and the computational demand low, a number of altitude shells are used to describe the LEO region in the range of 300 to 2000 km. Intact bodies and fragments are also grouped into diameter classes starting from 10 cm to 100 m. Similar approaches have been explored before. However, in this new model more source and sink mechanisms are considered as well as extra eccentricity classes. The NASA breakup model is used to estimate the rate of fragments caused by collisions and explosions (Johnson et al., 2001). The formulation of the equations is kept generic in the sense that the user of SANE can decide how many shells and classes are to be used in the simulation. An introduction to the model and preliminary results have been shown in Kebschull et al. (2013). While some changes to the model and its implementation will be shown here, a detailed look into the handling of collisions and explosions is given in a separate paper. Nevertheless a quick overview of the model will be given in Sec. 2. Its implementation is covered in Sec. 3. SANE is able to derive the flux based on the predicted number of objects in a given altitude shell. Based on the flux the environmental criticality can be derived. This will be discussed in Sec. 2.1.

*Corresponding author
Email address: c.kebschull@tu-braunschweig.de (Christopher Kebschull)

Nomenclature

Latin symbols:

A	cross sectional area
ADR	rate of actively removed objects
C	Environmental Criticality
N	number of objects
$\dot{N}$	rate of number of objects
$\bar{N}$	discrete rate of number of objects
PMD	post mission disposal rate
d	diameter
h	altitude
m	mass
n	maximum number of classes/shells
p	probability
Greek symbols:	
Δt	time step
ε	eccentricity
Φ	collision flux
Indices:	
I	intact bodies
F	fragments
c	circular orbit
col	collision
e	eccentric orbit
exp	explosion
i	shell/bin/class counter
p	perigee

2. Analytical Approach

In the model the LEO region is divided into altitude shells h_i. These shells are further subdivided into diameter bins d_i and eccentricity bins ε_i. An orbit is considered to be eccentric if the difference between perigee and apogee altitude is greater than the defined size of an altitude shell, i. e. the object would pass through multiple altitude shells. If an object is eccentric, it is equally distributed with respect to the altitude shells it passes. Due to the chosen altitude range the eccentricity span is defined between 0.0 and 0.135. At each instant of time t the number of objects for a given altitude shell h_i, a diameter class d_i and eccentricity class ε_i is given via:

$$N = f(d_i, h_i, \varepsilon, t) = N_{I_c} + N_{I_e} + N_{F_c} + N_{F_e}. \quad (1)$$

The number of objects N is given for intact bodies (N_I) and fragments (N_F), each on circular (subscript c) and eccentric (subscript e) orbits. The modification of the environment between two time steps is expressed as a differential equation:

$$\dot{N} = \frac{dN}{dt} = \frac{dN_{I_c}}{dt} + \frac{dN_{I_e}}{dt} + \frac{dN_{F_c}}{dt} + \frac{dN_{F_e}}{dt}. \quad (2)$$

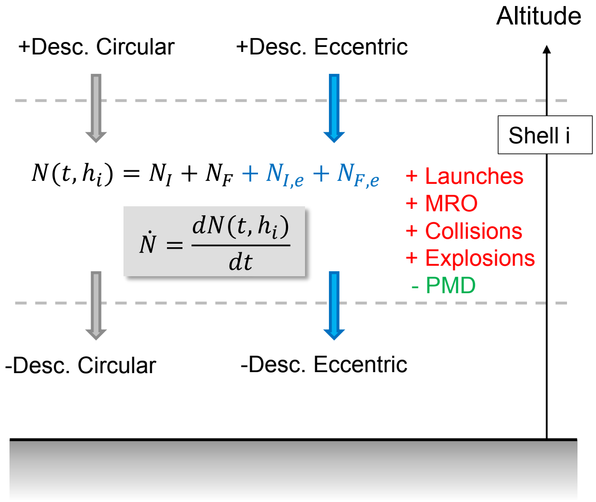

Figure 1: The source and sink mechanisms.

The Euler method is then used to determine the new number of objects for the next step:

$$N = N_0 + \dot{N} \cdot \Delta t. \quad (3)$$

The initial state of the simulation is defined by N_0. The most recent reference epoch in MASTER-2009 is used as the initial population. The step size Δt can be variable but in the context of the study it has been set to one year. The change rate for intact bodies is defined as:

$$\dot{N}_I = \frac{dN_I}{dt} = L + \bar{N}_{I_c} + \bar{N}_{I_e} - ADR - PMD - \bar{N}_{I_{col}} - \bar{N}_{I_{exp}}, \quad (4)$$

It is influenced by launches L, decay on circular ($\bar{N}_{I_c}$) and eccentric orbits ($\bar{N}_{I_e}$). Intact bodies can be removed via ADR and PMD maneuvers as well as fragmentations ($\bar{N}_{I_{col}}$, $\bar{N}_{I_{exp}}$). Fragmentations are considered as continuous source of fragment generation. An intact body may be removed while multiple fragments are created. Fig. 1 shows the basic principle of the model. A more detailed look into the model is given in Kebschull et al. (2013).

2.1. Environmental Criticality

The environmental criticality is a way of expressing the influence an object (e.g. satellite or rocket body) has on the space debris environment. Two proxies are used to factor in the risk (c_{risk}) that affects this object on its orbit and the impact (c_{impact}) it has on its environment in the case of a fragmentation event:

$$C_{crit} = c_{risk} \cdot c_{impact}. \quad (5)$$

In order to retrieve a value for the impact a fragmentation has on the environment, the change of the collision rate in every available cell (combination of altitude, eccentricity, diameter classes) of SANE's population is regarded.

Furthermore the fragmentation event is triggered in every time step of the simulation and the effects from that point on to the end of the simulation is considered. This leads to the following cumulative relation for a simulation over a given time frame, as it has been implemented in SANE:

$$C_{crit} = \sum_{t=t_{start}}^{t_{end}} \left[c_{risk,t} \cdot \int_{t_{frag}}^{t_{end}} (\Delta p) \, d\tau \right]. \quad (6)$$

Where $c_{risk,t}$ is defined is as

$$c_{risk,t} = \Phi \cdot A \cdot t, \quad (7)$$

with Φ as the collision flux, A as the cross sectional area of the target object, t the elapsed time of the simulation and

$$\Delta p = p_{frag} - p_{no\ frag} = (\Phi_{frag} - \Phi_{no\ frag}) \cdot A \cdot t \quad (8)$$

as part of the c_{impact} component. Again with Φ as the collision flux for the basic scenario in which no fragmentation has occurred and the current scenario where the target object has been fragmented and its fragments have spread over multiple shells and bins, which can be up to 27 336 cells (combination of 34 altitude shells, 12 diameter bins, 67 eccentricity classes). The risk portion of the equation ($c_{risk,t}$) considers the collision rate of the target object for each time step of the simulation time. The impact expressed as an integral over the time frame from the beginning of the fragmentation to the end of the simulation, determines the change in the overall collision rate. Fig. 2 illustrates this approach. The impact on the environment is determined in each time step of the simulation by fragmenting the target object and distributing its fragments over the affected cells. This creates a new initial snapshot of the space debris environment, which is used by the Population Generator to forecast the space debris environment based on this new version of the population. From the time of the fragmentation to the end of the simulation the difference in the collision rate in every cell is determined (Δp) by comparing the new version of the population affected by the fragmentation against the baseline population.

The reason for creating this fast analytical approach to forecast the space debris environment lies within the formulation of the environmental criticality, which when executed puts a high demand on the processing power that is needed.

3. Implementation

SANE has been implemented as a CLI (Command Line Interface) program using the programming language FORTRAN. This prototype has been designed so that the user has the ability to control as many options as possible. This has been realized by using configuration files that are read during the initialization phase. The main configuration file

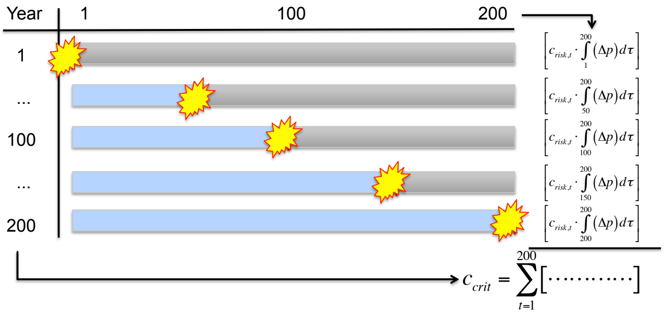

Figure 2: The environmental criticality respects the influence a fragmentation has on every region in LEO. A target object is fragmented in every possible epoch and its influence is analyzed by comparing the collision rate before and after the collision in every succeeding year (Δp). The risk of fragmentation is also part of the equation, determining the probability of the fragmentation just before it is triggered.

(config.inp) holds values for the simulation time frame and gives the user control over the fitting parameters that have been built into the model. Additionally the user can specify the mode in which SANE shall work. The first mode executes the Population Generator only. The second mode calculates the criticality of a target object. The Population Generator forecasts the space debris environment based on the formulation of the model into the future. This ability is essential for the second mode. The Criticality Computation relies on the comparison between a fragmentation and a non-fragmentation scenario.

There have been changes in the implementation compared to a previous version of the software. The residence time of an object in an altitude shell and eccentricity bin is no longer calculated using a simple atmospheric model. Instead a complex lookup table is used, which holds the residence per altitude shell and eccentricity bin in days. This way the decay behavior of the model can be changed fairly quickly by replacing the tables, which are stored as plain text files.

3.1. Population Generation

The Population Generator has the task to forecast the space debris environment. This is done using the analytical approach partly described above, which handles two kinds of objects, intact bodies and fragments. These objects loose their individuality once they have been placed into their corresponding cells. The source and sink mechanisms are implemented as separate modules and they are applied on these object clouds grouped into the cells. Each module can be activated and deactivated based on the requirement of the calculation. Fig. 3 shows how the population calculation is handled in SANE. After the initialization reads the configuration files (gray box), a loop over the simulation time frame determines the change of the population for a given time step of e.g. 1 year. The order of the modules is important, starting first with the launches, a source mechanism (all sources are red boxes) that simulates the launches of satellites and rocket bodies into various cells based on an 8 year repeating pattern. It is the assumption that an average lifetime of a mission is 8 years. Each satellite at its end of life is replaced by a new satellite. Following is the consideration of PMD and ADR maneuvers (all sinks mechanism are green boxes). After potential target objects have been (re-)moved, collisions and explosions are calculated. These two modules demand the most processing power. For this reason the collision module has been parallelized already. Due to the chosen analytical approach the individual collision groups (The model considers collisions due to pairing of altitude shells, eccentricity bins and diameter classes rather than individual object crossing analysis to determine the collision rate.) can be treated independent from one another, which leads to a good scaling of the model on multiple CPU cores and good speedup of the calculation. At the end of the process chain the natural decay of the objects is considered. This mechanism moves the object groups through the altitude and eccentricity shells, removing objects that are close the Earth's atmosphere (lower than 300 km) from the simulation eventually. After each time step has been processed the content of each of the 27 336 cells is stored. At the same time it serves as the starting point for the following simulation year. After the simulation ends the data is prepared for output, which can then be used for further analysis or the stored snapshots are used in the following Criticality Computation. The runtime of the Population Generator for a 200 year simulation is at about 120 seconds on a 2.2 Ghz Intel i5 mobile processor with 4 threads available.

3.1.1. Initial Population

The simulation has a defined starting point, which is the reference epoch May 1., 2009 as provided in MASTER-2009. In this version of SANE the eccentricity is regarded, which leads to an updated initial population in regard to the previous publication (Kebschull et al. (2013)). As before, the population has been divided into two categories of objects, intact bodies and fragments. They are distinguished in the model by different area-to-mass ratios. Fragments have a ratio of 0.36 m^2/kg, while intact bodies are assumed to have a ratio of 0.005 m^2/kg. These values have been derived in an internal study at the ILR. The objects are then distributed among the three cell types, altitude, diameter and eccentricity.

The objects of each category per altitude shell are shown in Fig. 4. High density areas of fragments and intact bodies are visible at about 800 km and 1400 km altitude. For this simulation 34 altitude shells have been defined, each 50 km in size.

Fig. 5 shows the distribution of the population among the 67 eccentricity shells. Based on the limitations of the model, which sets the altitude range from 300 km to 2000 km, a maximum eccentricity value of about 0.12 is set.

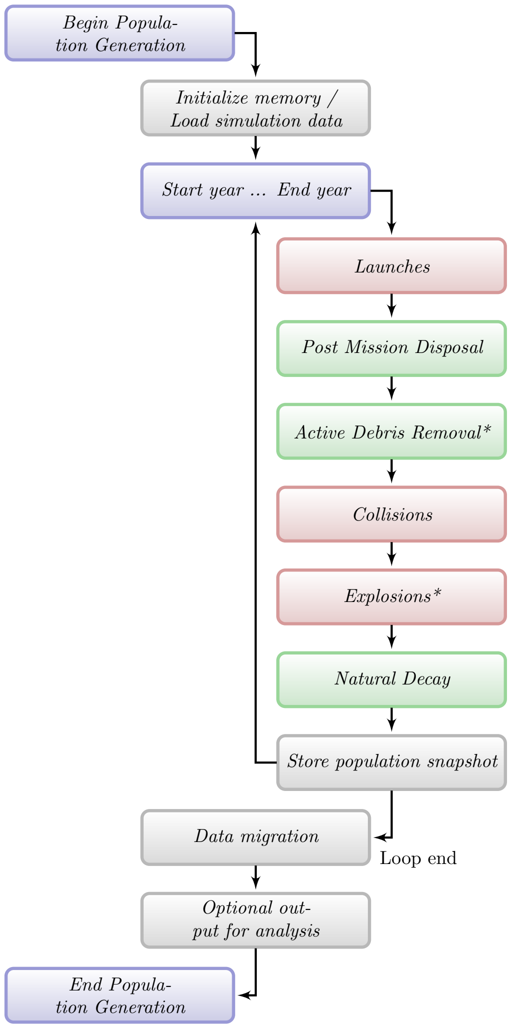

Figure 3: Details of the process for generating the space debris population needed for the entire simulation span. *Explosions and mitigation measures have not been validated yet.

A majority of the objects in the initial population are on nearly circular orbits.

For the distribution of the objects among the 12 diameter bins a power law has been derived, creating the diameter bins. The lower limit is set to 10 cm, the upper limit is at 100 m:

$$d_i[m] = 10^{(0.25 \cdot i - 1)}. \tag{9}$$

The distribution of the the intact bodies and fragments is shown in Fig. 6. The first three bins are dominated by fragments. Larger objects are usually intact bodies.

The population is loaded in the initialization phase and distributed among the cells. It can also be modified by the user by replacing the file. Thus a new version of the future space debris environment can be created.

3.1.2. Population Validation

Through the main configuration file the user can control to write population snapshots as plain text files to the

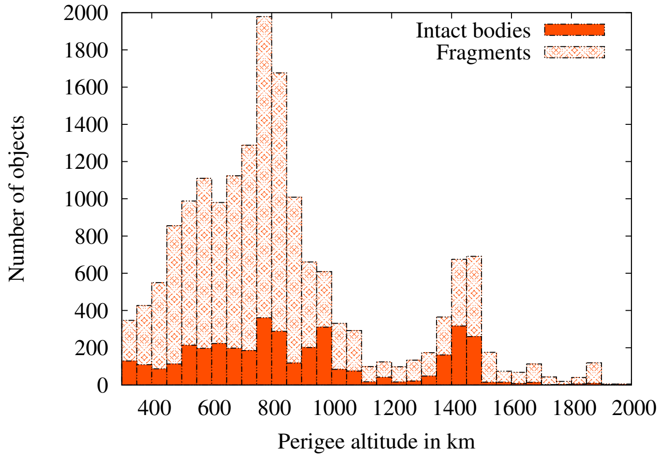

Figure 4: Objects of the initial population distributed into altitude shells.

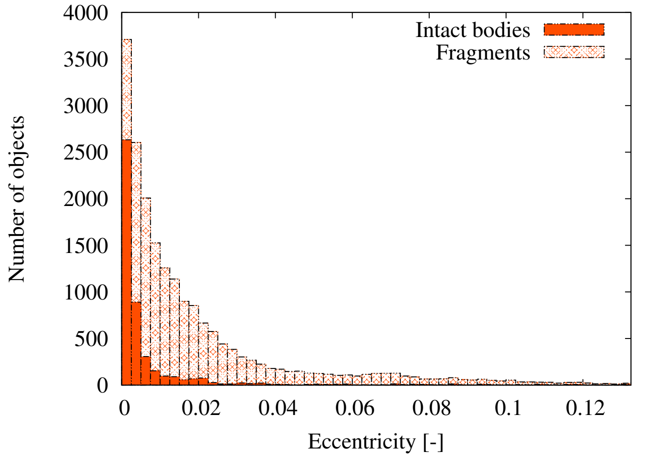

Figure 5: Objects of the initial population distributed into eccentricity bins.

output folder. With this data the generated population can be analyzed and validated against other tools. In the current state SANE has been fitted against results from the numerical tool LUCA (**L**ong-term **U**tility for **C**ollision **A**nalysis). As mentioned before each source and sink module can be activated and deactivated in the configuration file. This has been used to fit each mechanism individually. Once this had been accomplished the combinations of the sources and sinks have been looked at closely, to see whether the dynamical behavior of the model shows unwanted effects. This has especially been useful to validate the collision module, which is highly sensitive to even slight variations in the population. The results for the validation of the combination of the launches and decay modules are shown in Fig. 7. Snapshots of the years 2010, 2100 and 2200 are shown from top to bottom. Only the intact bodies over the altitude are analyzed. The forecast by LUCA is shown as a solid red line, while the results by SANE are shown as a dashed green line. Both modules show characteristic peaks. After the first time step SANE

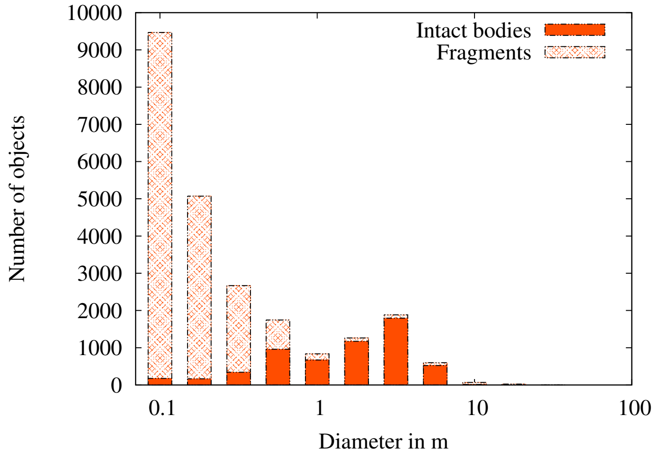

Figure 6: Objects of the initial population distributed into diameter bins.

creates a population that is very close to the one generated by LUCA. For the year 2100 the allover trend still looks OK, however a trend becomes obvious where the decay might be too high for the objects around 1400 km altitude, pushing to many objects to lower orbits. This leads to an overestimation in the shells between 1200 - 1400 km, while 1400 - 1500 km are underestimated. This effect becomes even more obvious in the year 2200. However the allover trend looks very good, nothing that would suggest a flaw in the underlying model. There are two ways for overcoming the shortcoming in the 1400 - 1500 km altitude shell. First the fitting parameter can be updated in the implementation so that it is not a simple scaling variable but rather a function of the altitude. This way the user could update the configuration file and retrieve a different result. The preferable way to approach this issue however would be to update the lookup table for the decay in the given region. The residence times for objects in the altitude shells in question have to be increased.

The fragment population has already been validated. The modules for launch, decay and collisions have been activated to retrieve the results shown in Fig. 8. Again snapshots from 2010, 2100 and 2200 have been analyzed. Both LUCA and SANE show the same characteristic peaks at 800 and 1400 km altitude. In the year 2010 SANE shows a slight underestimation of fragments in the 750 km shell, while the 700 km altitude shell is slightly overestimated. The same effect is visible for the 1400 and 1450 km altitude shells. For the year 2100 the small under- and overestimations move to different shells and seem to stretch out over the entire orbital region. For the year 2200 the forecasts from both tools look very similar. SANE underestimates LUCA slightly in the high altitude region from 1500 km to 2000 km. In the high density area between 700 and 800 km SANE overestimates the number of fragments by about 6%.

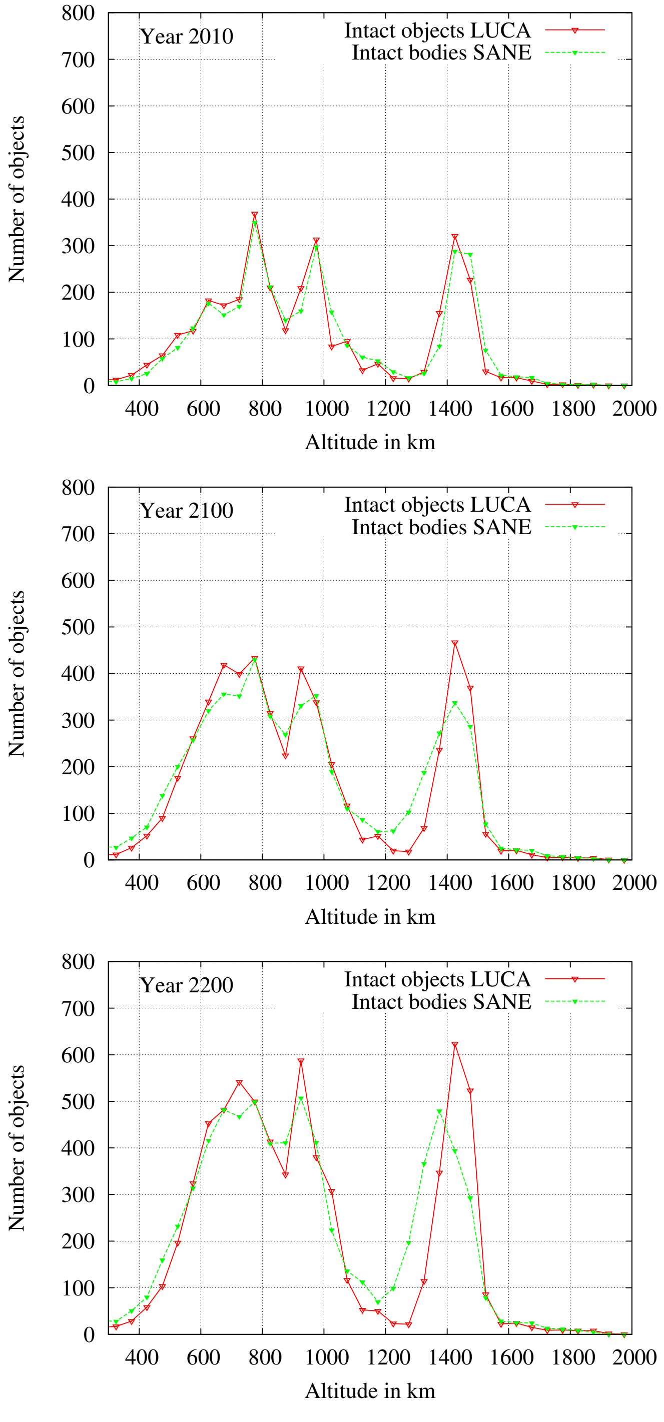

Figure 7: Evolution of intact bodies over time when considering launches and decay in SANE only in comparison to results from LUCA for the years 2010, 2100 and 2200.

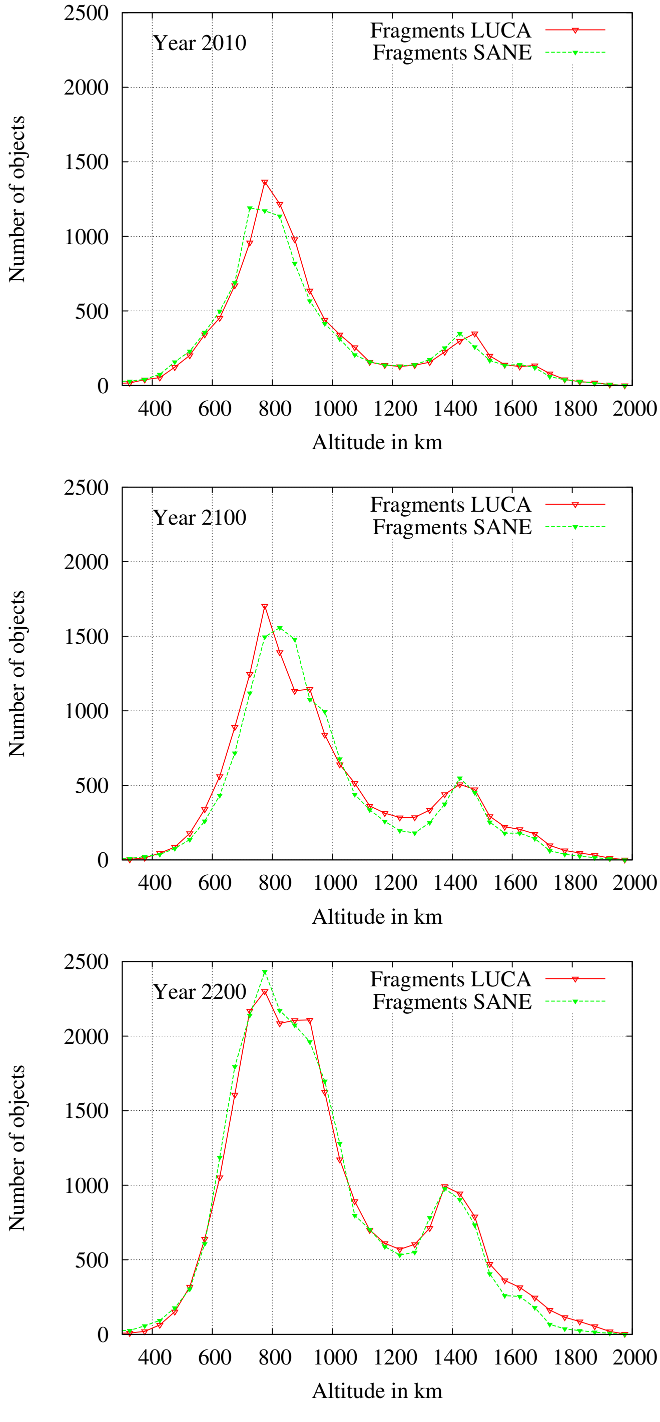

Figure 8: Evolution of fragments over time when considering launches, decay as well as collisions in SANE in comparison to results from LUCA for the years 2010, 2100 and 2200.

3.1.3. Collision Flux

The main motivation for generating the forecast of the future space debris environment is to retrieve the collision flux. A linear relation has been found to estimate the flux based on the number of objects in a given altitude shell:

$$\Phi = f(h_i, N(h_i)) = \alpha_1(h_i) \cdot N(h_i) + \alpha_0(h_i). \qquad (10)$$

This relation has been derived based on the analysis of the reference epochs 1990, 1996, 2001, 2005 and 2009 in MASTER-2009. For each altitude shell the parameters α_0 and α_1 have been stored in a lookup table, which is also read in the initialization phase. The parameters for the flux have been retrieved for sun-synchronous orbits only. Thus, flux estimates can be considered as being quite conservative. In future versions different inclination bins can also be considered. Based on this relation the collision flux on target orbits can be derived. The details of the target objects can be specified in the main configuration file. In the current state SANE assumes that the objects do not have any maneuvering capability, so they are not keeping a constant altitude, which would be the case for active satellites during their mission time. Thus each target object is also subject to natural decay in the simulation time frame. Tab. 1 shows four reference objects on different orbits. These objects have been analyzed using SANE over a simulation time frame of 200 years. The results of the target object simulations can be seen in Fig. 9. The flux estimation by SANE is shown as green triangles, while LUCA's results are shown as solid red line and triangle points. For the results by LUCA the standard deviation of the flux is also visible as dashed line with stars as points. The first object has an inclination of 63.4°. Over the entire simulation time an over estimation of the flux can be observed, even above the upper standard deviation line. The estimated flux of object no. 2 moves close to LUCA's estimate and within the upper and lower standard deviation bounds. Towards the end the trends of both results seem to deviate. SANE begins to overestimate again. The third object is at an inclination of 85°. The flux over time estimation by SANE follows the same trend as the one created by LUCA. There is very little deviation visible. Object no. 4 is on a sun-synchronous orbit. The result by SANE shows slight over estimation but the result is still within the bounds of the standard deviation.

Table 1: Overview of reference objects used as targets in the flux and Criticality Computation.

ID	Mass [kg]	SMA [km]	Ecc. [-]	Inc. [deg]
Obj 1	919.0	7484.0	3.9E-02	63.4
Obj 2	2131.0	7596.4	1.5E-03	74.3
Obj 3	2725.0	7167.5	1.9E-03	85.0
Obj 4	980.5	7170.4	1.3E-03	98.5

In Tab. 2 the relative deviation between both forecasts of the collision flux for each reference object is analyzed.

Table 2: Average relative flux deviation from the value derived by LUCA.

ID	Average deviation [%]
Obj 1	27.69
Obj 2	18.59
Obj 3	3.45
Obj 4	9.38

The deviation in the flux arises from the parameters for the linear relation of the collision flux, Eq. 10. The parameters have been derived for sun-synchronous orbits only. When applying the relation on lower inclinations an over estimation can occur (as seen for Obj 1 and Obj 2), due to the fact that the allover flux on lower inclination

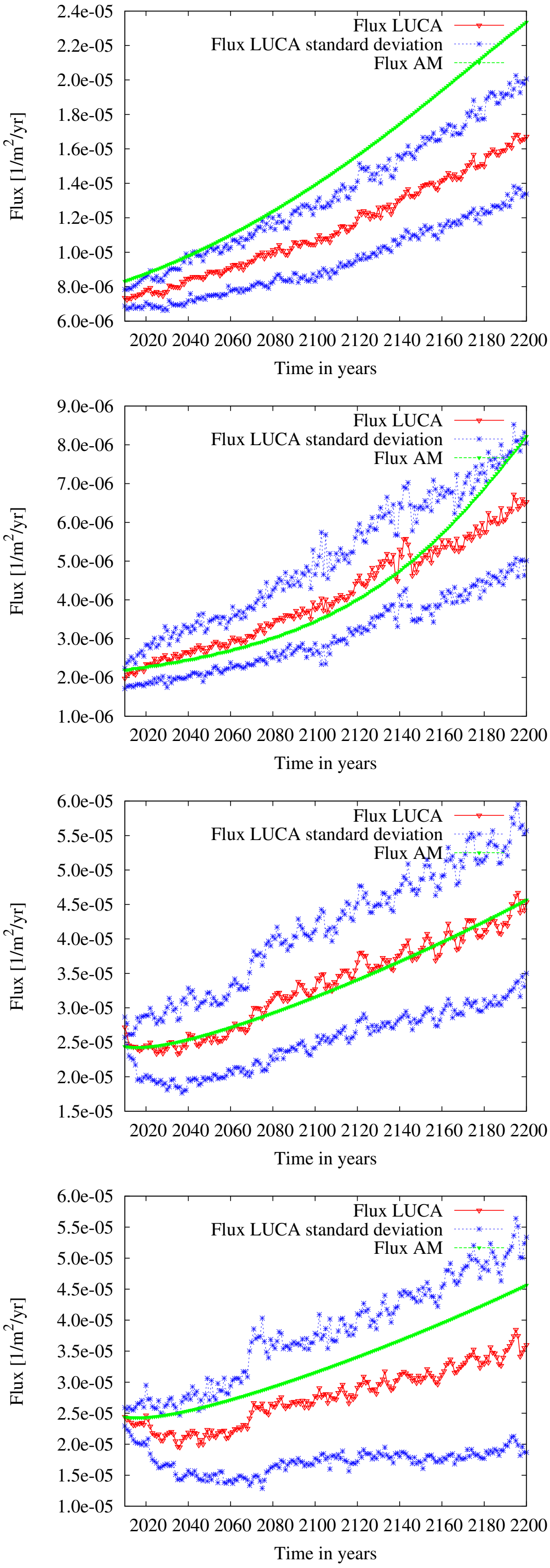

Figure 9: Collision flux of reference objects in 63° to 65° (1), 71° to 74°(2), 82° to 86° (3) and 98° to 99°(4) inclination bins compared between LUCA and SANE.

is lower. A way optimize the flux estimation is the formulation of multiple inclination bins when considering target objects.

3.2. Criticality Computation

The second operation mode of SANE computes the environmental criticality of a given target object. The described process of deriving the future population is an integral part of the criticality computation. As shown in Fig. 10 the Population Generator (green) is used once initially to create the non-fragmentation population for the entire simulation time span. Once it has been generated the simulation starts and loads the first snapshot for the epoch. Within this epoch the target object is fragmented (gray). The second layer loop computes the impact the fragmentation has (expressed through the criticality) from the time of the fragmentation to the end of the simulation time span. Within this loop the Population Generator is used again to forecast the environment with the new fragments that have been added. The criticality that is derived for the given epoch (red) is based on the collision rate difference in every cell of the environment (altitude, diameter, eccentricity) between the just created environment with fragments and the initially computed non-fragmentation environment.

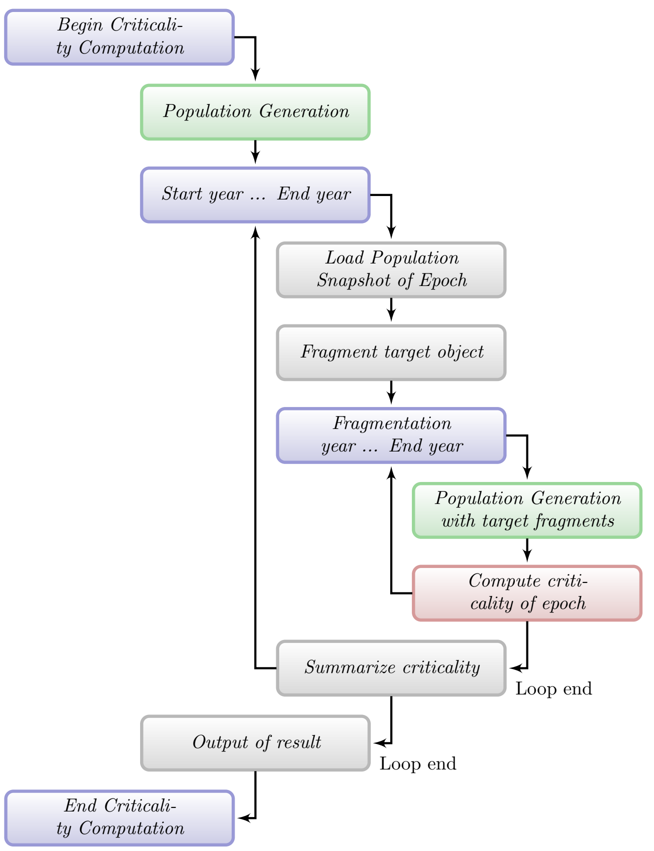

Figure 10: Details of the process for computing the environmental criticality as needed for the entire simulation span.

For the reference objects in Tab. 1 the criticality has been computed. Tab. 3 shows the criticality for 25, 50, 100 and 200 year simulation time frames. Because of the cumulative nature of the criticality definition the values increase for larger simulation time spans (Unless an object entered the Earth's atmosphere and thus has no longer an impact on the environment. The criticality will remain the same in this case, no matter how long the simulation time frame is.). In the current case the criticality values span over 5 orders of magnitude.

Table 3: Criticality results for different simulation time frames.

ID	25 yrs [-]	50 yrs [-]	100 yrs [-]	200 yrs [-]
Obj 1	0.55E-05	0.19E-04	0.78E-04	0.25E-03
Obj 2	0.14E-03	0.89E-03	0.71E-02	0.50E-01
Obj 3	0.21E-03	0.68E-03	0.22E-02	0.34E-02
Obj 4	0.29E-04	0.95E-04	0.31E-03	0.44E-03

Fig. 11 shows the evolution of the criticality value on a logarithmic scale. At the end of the simulation Obj 2 (dashed green line) has the highest criticality value among the reference objects, being about 15 times more "critical" than the next object on the list (Obj 3, dotted blue line), which has a constant criticality value between the years 2165 and 2209. For short simulation time frames, like 25 years, Obj 3 would lead the list followed by Obj 2. Both objects swap positions around the year 2050. The reason for this development is the decay of both objects. While Obj 3 is on a lower altitude with higher flux it quickly moves into less populated regions, where its risk of being hit is lower and also the impact of its fragmentation is not as crucial. Obj 2 on the other hand started out at a higher altitude where the flux was lower and thus its lower risk of being fragmented caused a smaller criticality value. With the simulation time advancing Obj 2 slowly decayed into more populated regions, increasing its criticality value over Obj 3's. Both objects have a mass between 2 and 3 tons. Obj 1 (solid red line) and 4 (double dotted violet line) are on position 4 and 3 respectively. At the end of the simulation the trend of Obj 1 suggests that at a longer simulation time frame it would swap places with Obj 4, which has re-entered into Earth's atmosphere around the year 2160 and thus has a stagnant criticality value.

The environmental criticality is a time dependent value, which does not solely rely on the mass or initial orbit of an object but also takes into account the evolution of the target orbit. This leads to different results when different simulation time spans are considered. While a higher mass causes a greater impact on the environment, when a fragmentation occurs, the target orbit has a big part in the evolution of the criticality.

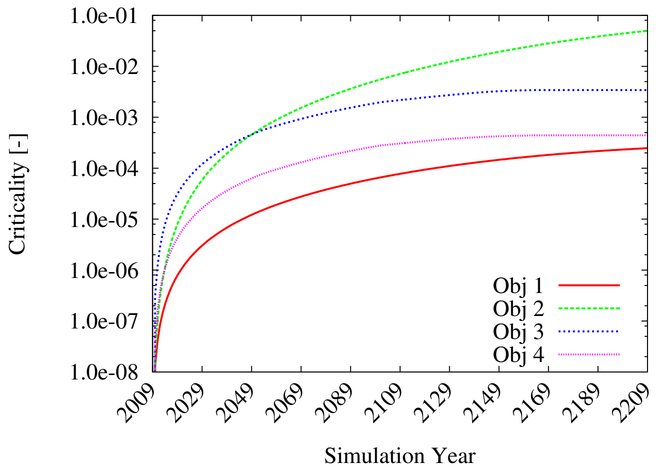

Figure 11: The evolution of the criticality of a selected group of reference objects on a logarithmic scale over the time span of 200 years.

4. Conclusion

In this paper the implementation of an analytical model to forecast the evolution of the space debris environment, named SANE (**S**imple **AN**alytical **E**nvironment) has been introduced. Its internal processes of the two operational modes (Population Generator and Criticality Computation) have been outlined. The implementation of the model has been kept flexible by considering the fitting parameters that have been designed into the equation so to give the user control over the model behavior. Additionally many input values like the residence time, or the collision flux parameters are supplied based on lookup tables in plain text format. These files can easily be replaced. Each parameter has a big influence on how the simulation plays out. The output of the simulation gives the user an overview of the population and its development after the simulation time ends (number of intact bodies and fragments) from which the collision flux can be derived. The collision flux is essential for the calculation of the environmental criticality, which has also been introduced as a value that carries information about the risk a target object is subjected to combined with the impact it has on the space debris environment in the case of its fragmentation. The validation of the population forecast has been performed by comparing the results to a numerical tool called LUCA (**L**ong-term **U**tility for **C**ollision **A**nalysis). SANE was able to generate good results in reproducing the trend in the altitude shells in LEO for the modules launches, decay and cdecayollisions. The collision flux per altitude shell can be derived based on the number of objects in the given shell. MASTER-2009 reference epochs have been used to derive an suitable linear expression for this matter. The population forecast and the derived relation were good enough so that the collisions flux on target objects could be reproduced with an average deviation of about 4 - 28 %. Based on the collision flux in turn the environmental criticality can be computed. Results for 4 reference objects have been shown. It could be determined that not only the mass of an object is a driving factor for the environmental criticality but also the simulation time and the orbit of the target object. The evolution of the environmental criticality has been shown. In future campaigns SANE can be used to derive new priority lists based on the environmental criticality of target objects as the criterion for the ranking system. Additionally SANE can be further developed by adding flux bins for more orbits than just sun-synchronous orbits. This would lead to more accurate forecasts for the collision flux on target orbits. Also in the current implementation no interpolation has been regarded when moving between the cells. When considering the movement of a target object by the means of a more precise propagator instead of moving it, like the rest of the objects in the model, through the shells and bins, an interpolation technique would lead to a more smooth transition between the cells e.g. when the object moves from one altitude shell to the next. SANE can also be extended by a generic propagator interface that has been developed at the ILR (Thomsen (2013), Möckel et al. (2013)). With this interface different types of propagators can be used, analytical, semi-analytical, numerical or parallelized hybrids that already work on GPUs, which can lead to a good speedup for a large number of objects (Möckel et al., 2012). In future studies the effect of different kinds of long-term ADR and PMD scenarios can be investigated with a reduced computational effort by no longer relying on complex numerical Monte-Carlo simulations. Not only the impact on the environment but also the pros and cons of the different strategies like single or multiple target missions or the application of electric or chemical propulsion systems can be of interest in this context. The information can be used in conjunction with a cost estimation model, which will be discussed in a separate paper published later this year, that is based on Wiedemann et al. (2012) and Braun et al. (2013). As a result cost effective ADR scenarios can be derived using a cost model plugin combined with SANE. Additionally SANE's sensitivity toward changes of the boundary conditions can be investigated e.g. by choosing an alternative launch pattern or different collision and explosion parameters. The impact of those variations on the resulting population, collision flux and environmental criticality is of interest. Furthermore the behavior of the environmental criticality can be looked at for longer time frames and more target objects. The influence of the risk and the impact component of the equation can be analyzed in more detail. This could lead to a recommendation for an optimized simulation runtime to retrieve a priority list or a statement of a "criticality" value for any given object in the context of a reference list.

5. Acknowledgments

The development of the described analytical model is a part of the project *Fragmentation Consequence Analysis for LEO and GEO Orbits* supported by ESA's General Studies Programme (contract 4000106517/12/F/MOS).

6. References

H. G. Lewis, G. G. Swinerd, R. J. Newland, and A. Saunders, "The fast debris evolution model," *Advances in Space Research*, vol. 44, pp. 568–578, 2009.

A. Rossi, A. Cordelli, P. Farinella, and L. Anselmo, "Collisional evolution of the earth's orbital debris cloud," *Journal of Geophysical Research*, vol. 99, no. E11, pp. 23,195–23,210, November 1994.

J. Radtke, D. Flegel, J. Gelhaus, M. Möckel, V. Braun, C. Kebschull, C. Wiedemann, H. Krag, K. Merz, and P. Vörsmann, "Revision of statistical collision analysis for objects inside of satellite constellations," no. IAC-13,A6,P,8.p1x18951. 64th International Astronautical Congress 2013, September 2013.

N. L. Johnson, P. H. Krisko, J.-C. Liou, and P. D. Am-Meador, "Nasa's new breakup model of evolve 4.0," *Advances in Space Research*, vol. 28, no. 9, pp. 1377–1384, 2001.

C. Kebschull, V. Braun, S. Flegel, B. Reihs, S. Hesselbach, J. Gelhaus, M. Möckel, J. Radtke, C. Wiedemann, H. Krag, and P. Vörsmann, "A simplified approach to analyze the space debris evolution in the low earth orbit," no. IAC-13,A6,2,3x19121. 64th International Astronautical Congress, September 2013.

P. Thomsen, "Gpu-basierte analyse von kollisionen im weltraum," Master's thesis, TU-Braunschweig, September 2013.

M. Möckel, T. Thomsen, C. Kebschull, S. Flegel, J. Gelhaus, V. Braun, J. Radtke, C. Wiedemann, and P. Vörsmann, "A fast, modular approach to object propagation and collision analysis," no. IAC-13,A6,P,14.p1,x18052. Internation Astronautical Congress, 2013.

M. Möckel, C. Kebschull, S. Flegel, J. Gelhaus, V. Braun, C. Wiedemann, and P. Vörsmann, "Flexible implementation of orbital propagators in heterogenous computing environments." 61. Deutscher Luft- und Raumfahrtkongress, September 2012.

C. Wiedemann, S. Flegel, M. Möckel, J. Gelhaus, V. Braun, C. Kebschull, J. Kreisel, M. Metz, and P. Vörsmann, "Cost estimation of active debris removal," no. IAC-12.A6.5.3. Internation Astronautical Congress, 2012.

V. Braun, A. Lüpken, S. Flegel, J. Gelhaus, M. Möckel, C. Kebschull, C. Wiedemann, and P. Vörsmann, "Active debris removal of multiple priority targets," *Advances in Space Research*, vol. 51, pp. 1638–1648, 2013.